Besides adding foreign key constraints at creation time (i.e. when using `CREATE TABLE`), you can also add such constraints after a table has been created with help of `ALTER TABLE`. You can also remove foreign key constraints from an existing table.

Adding Foreign Key Constraints Via ALTER TABLE

MySQL & Postgresql

```
ALTER TABLE <table_name>
ADD FOREIGN KEY <column_name> REFERENCES <related_table> (<related_column>) ON DELETE ... ON UPDATE...
```

Removing Foreign Key Constraints Via ALTER TABLE

MySQL & Postgresql

```
ALTER TABLE <table_name>
DROP FOREIGN KEY <constraint_name>;
```

In order to DROP a foreign key constraint (just as for dropping any other kind of constraint), you need to assign a name to the constraint when adding it.

This can be done like this:

Adding with constraint name via CREATE TABLE:

```
CREATE TABLE <table_name> (
  <column_name> <column_data_type> FOREIGN KEY <constraint_name> REFERENCES ...
);
```

The `FOREIGN KEY` part (i.e. `FOREIGN KEY <constraint_name>`) is only required if a name should be assigned. If you don't need a name (because you don't plan on dropping the constraint at any time), you can omit `FOREIGN KEY` - just as we did in the lectures.

There also are ways of dropping constraints without names but it's a bit more complicated - see: https://stackoverflow.com/questions/1430456/how-to-drop-sql-default-constraint-without-knowing-its-name

Adding with constraint name via ALTER TABLE:

```
ALTER TABLE <table_name>
ADD CONSTRAINT <constraint_name> FOREIGN KEY <column_name> REFERENCES ... (see above)
```